

Article

Detection of Road Surface Changes from Multi-Temporal Unmanned Aerial Vehicle Images Using a Convolutional Siamese Network

Truong Linh Nguyen¹ and DongYeob Han^{2,*}

- ¹ Faculty of Information Technology, Hanoi University of Mining and Geology, No. 18 Pho Vien, Duc Thang, Bac Tu Liem, Ha Noi 10000, Vietnam; linhgeo89@gmail.com
- ² Department of Civil Engineering, Chonnam National University, 77 Yongbongro, Bukgu, Gwangju 61186, Korea
- * Correspondence: hozilla@chonnam.ac.kr

Received: 12 February 2020; Accepted: 19 March 2020; Published: 22 March 2020



Abstract: Road quality commonly decreases due to aging and deterioration of road surfaces. As the number of roads that need to be surveyed increases, general maintenance—particularly surveillance—can be quite costly if carried out using traditional methods. Therefore, using unmanned aerial vehicles (UAVs) and deep learning to detect changes via surveys is a promising strategy. This study proposes a method for detecting changes on road surfaces using pairs of UAV images captured at different times. First, a convolutional Siamese network is introduced to extract the features of an image pair and a Euclidean distance function is applied to calculate the distance between two features. Then, a contrastive loss function is used to enlarge the distance between changed feature pairs and reduce the distance between unchanged feature pairs. Finally, the initial change map is improved based on the preliminary differences between the two input images. Our experimental results confirm the effectiveness of this approach.

Keywords: change detection; convolutional Siamese network; unmanned aerial vehicles; image processing

1. Introduction

The quality of road surfaces will decrease during use due to aging and deterioration. Some damage will always appear on a road surface, such as potholes and cracks—the two most common categories of road surface damage. To ensure safety for traffic, maintaining road surface quality is both necessary and urgent. The number of roads to survey is increasing, which poses a real challenge for managers using traditional surveying methods as it leads to increasing costs. Usually, an inspector needs to go outside to collect information about the position and condition of the surface, then plan to repair the damaged location. Currently, the use of an unmanned aerial vehicle (UAV) supported by a high-level computing device and artificial neural network makes this surveying aspect more efficient and more cost-effective than traditional methods.

Image change detection aims to detect the changed areas in images of the same scene taken at different points in time [1,2]. Over the last three decades, many different methods have been reported for detecting a changing area [3–7]. Alcantarilla et al. proposed a novel approach to change detection in Google Street View using monocular video sequences [8]. The method combines geometric methods with the learning made possible with an efficient convolutional network to discriminate between actual and nuisance changes. Guo et al. proposed a method based on convolutional neural network (CNN) architecture. It measures changes to a region using an implicitly learned metric, then develops a contrastive loss threshold to overcome noisy changes using a different viewpoint [9]. To detect temporal



changes in a scene from a pair of images, a new method that integrates CNN features with superpixel segmentation has been introduced [10]. Superpixel segmentation is integrated to estimate the precise segmentation boundaries of the changes. Nemmour and Chibani proposed that the combination of fuzzy sets and neural networks provides complete information on changes in a remotely sensed image [11]. A fuzzy membership model classifies multi-temporal images into changed and unchanged classes, and newly urbanized areas are detected based on an artificial neural network with the input of two Landsat Thematic Mapper images obtained at different times. The result of the method is effective, and detailed classes are created [12]. On the other hand, Wang et al. investigated the uncertainty in detecting the change of images. According to the authors, there is a need to be transparent in assessing that uncertainty. Therefore, they proposed a framework for evaluating binary land change utilizing remote sensing images. First, changed and unchanged classes are classified by two widely adopted image change detection methods. Second, binary decisions are reached through thresholding on change maps. Finally, two sampling designs (i.e., stratified sampling and random sampling) are used to evaluate the results [13].

There are also many studies on change detection in UAV images. Zhan et al. proposed a novel model for change detection in optical aerial images, which is based on the supervised deep Siamese CNN [14]. A multi-temporal change detection framework proposed by Song et al. covers changes to cultivated land in mountainous terrain [15]. The data in the paper, with very fine spatial and temporal resolutions, was collected by small UAVs. Shi et al. introduced an object-based method to detect change using multi-temporal images obtained by UAV [16]. This method can overcome distortion effects and can fully use the high resolution of UAV images [17]. Changes to urban areas in the city of Konya, Turkey are detected by finding the difference in digital elevation models based on comparisons of time-series point cloud data from aerial images taken at different times [17].

In this study, a change detection method considering road surface as a property was presented using high-resolution UAV images acquired for road surface inspection. First, a convolutional Siamese network (ConsimNet) was proposed to extract the features of image pairs, and a Euclidean distance function was applied to calculate the distance between two features. Then, the contrastive loss function [14,18] was used to pull the unchanged pairs together and push the changed pairs apart. Finally, an edge detection technique was applied to improve the detected area in the initial change map. The edge detection finds the boundary of changed areas, and the detected area in the initial change map is adjusted based on this boundary.

ConsimNet has proven effective at overcoming certain problems encountered in detecting changes in high-resolution images, such as differences in an object due to a different viewpoint, wrong detection due to the shadow of an object, and inaccurate geometric correction. However, the limitation of ConsimNet is that it can only detect areas of significant change; it is difficult to detect changes that are unclear or blurry. Of an entire changed area, ConsimNet often detects only the central part of the change region, neglecting the rest because the former is usually the area with the most major changes and thus has clearer differences than the surrounding region (the area in the process of being broken up). Therefore, the area of the change is not fully detected. In this study, edge detection was used to overcome this issue and ensure that such changes, specifically the boundary of the changing region, would be found. Using this range as a standard reference for the initial change map means it would be able to detect exact defects and the range of defects. With this method, the locations of the changing regions were identified while ensuring that the entire area of those regions was detected. This method can be used in pre-warnings of road conditions in road inspections, even if existing conditions are not bad. Small potholes, indentations, or other abnormal features of the road surface can be detected. The rest of this paper is structured as follows. The methodology is described in Section 2. The experiment is presented in Section 3. The results and discussion are shown in Section 4. Finally, the conclusion from this study is drawn in Section 5.

2. Methodology

The schematic of the proposed method is shown in Figure 1. A pair of images is input through a convolutional Siamese network (ConsimNet) [19] to obtain feature pairs. A simple predefined distance metric (Euclidean distance function— L_2) is then used to measure the dissimilarity of the feature pairs. The contrastive loss function is applied to bring together unchanged pairs and separate changed pairs. However, the initial change map results are not commensurate with the real changed area; the extent of the changing area is not fully detectable. To obtain full coverage, the boundary of the real changing area needs to be obtained as a reference value from which to adjust the range of the changing area.



Figure 1. Schematic of the proposed method: image pair is fed to the convolutional Siamese network (ConsimNet) to obtain feature pairs. After obtaining the dissimilarity of feature pairs, contrastive loss is applied to pull unchanged pairs together and push changed pairs apart; this is then used to improve the accuracy of the change map.

2.1. Convolutional Siamese Metric Network

Siamese networks are neural networks containing two or more identical sub-network components [19]. The networks have the same configuration, parameters, and weights. There are three types of layers in conventional CNNs: convolutional, pooling, and fully connected as demonstrated in Figure 2.



Figure 2. Siamese neural network architecture.

The convolutional layers can extract the hierarchical features from the input image. The functionalities of the pooling layers consist of receptive field enlargement and dimensionality reduction, which means to reduce the size of the output feature maps. The fully connected layers are used as a classifier, which outputs the probabilities predicting the input image to each class.

2.2. Contrastive Loss Function

The contrastive loss function was used to enlarge the distance between changed pairs and reduce the distance between unchanged pairs simultaneously. Let $X = \{x(i,j) | 1 \le i \le h, 1 \le j \le w\}$ be an aerial image, and X_1 and X_2 be two input images each with a size of $h \times w \times 3$, where w and h are spatial dimensions and 3 is the channel dimension (RGB channels). Define the parameterized distance function to be learned D_W between X_1 and X_2 as the Euclidean distance between the outputs of G_W :

$$D_{W}(X_{1}, X_{2})_{i,j} = \left\| G_{W}(X_{1})_{i,j} - G_{W}(X_{2})_{i,j} \right\|_{2}$$
(1)

 $G_W(X_1)$, $G_W(X_2)$ is the output vector tensor, $G_W(X_1)_{i,j}$, $G_W(X_1)_{i,j}$ is the feature vector of the pixel with location (i, j) in the image *X*. To shorten the notation, $D_W(X_1, X_2)_{i,j}$ is written as $D_{i,j}$. Then, the loss function in its most general form is:

$$l(W) = \sum_{k=1}^{P} L(W, (Y, X_1, X_2)^k) = \sum_{k=1}^{P} \sum_{i,j} (1 - y_{i,j}^k) L_U(D_{i,j}^k) + y_{i,j}^k L_C(D_{i,j}^k)$$
(2)

where *Y* is a binary ground-truth map assigned to the input image pair and y(i,j) = 0 if the corresponding pixel pair is deemed similar or y(i,j) = 1 if it is deemed dissimilar. L_C is the partial loss function for a pair of dissimilar points and L_U is the partial loss function for a pair of similar points.

 $L_{\rm C}$ and $L_{\rm U}$ must be designed such that minimizing *L* with respect to $D_{\rm i,j}$ produces a low value for a pair of unchanged pixels and a high value for a pair of changed pixels. $L_{\rm C}$ and $L_{\rm U}$ are defined as follows:

$$L_{\rm U}(D_{\rm i,j}^k) = \frac{1}{2} (D_{\rm i,j}^k)^2 \tag{3}$$

$$L_{\rm C}(D_{i,j}^k) = \frac{1}{2} \left\{ \max(0, m - D_{i,j}^k)^2 \right\}$$
(4)

where *m* is a margin. Change pixel pairs contribute to the loss function only if their parameterized distance is within this margin. In the experiment, *m* was set to 2. Thus, the final loss function is:

$$L(W, (Y, X_1, X_2)^k) = \sum_{i,j} (1 - y_{i,j}^k) \frac{1}{2} (D_{i,j}^k)^2 + y_{i,j}^k \frac{1}{2} \left\{ \max(0, m - D_{i,j}^k)^2 \right\}$$
(5)

2.3. Improvement of the Results

The purpose of this step is to find results that improve on the initial results based on the preliminary difference between the two input images. Therefore, it is necessary to find the boundaries of the areas of difference of the two images. This boundary is considered the standard range within which to expand the initial detected area in the previous step until touching the boundary.

The steps are as follows:

Step 1: Find the difference between two images I₁, I₂ in each RGB color channel:

$$\begin{split} & \text{Red}_{I_1-I_2} = \text{Red}_{I_1} - \text{Red}_{I_2} \\ & \text{Green}_{I_1-I_2} = \text{Green}_{I_1} - \text{Green}_{I_2} \\ & \text{Blue}_{I_1-I_2} = \text{Blue}_{I_1} - \text{Blue}_{I_2}. \end{split}$$

Step 2: Detect the edges of the two images I_1 , I_2 by Canny edge detection, and by combining this with the result in the above step we obtain a result such as that shown in Figure 3c.

Step 3: Group all the adjacent pixels, and fill into the group that has a closed pixel area (Figure 3d). Step 4: Remove the small region and determine the boundary location (Figure 3e,f). Step 5: Based on this boundary, expand the initial change map.



Figure 3. The boundary of the changed area was found: (a) image at time t_0 ; (b) image at time t_1 , (c) the changed area between the two images; (d) after filling holes in the closed pixel area; (e) after removing the small region; and (f) the locations of different areas were detected.

Figure 3f is the preliminary difference between the two input images (white pixels). The result not only includes real changes (red box) with the whole area covered, but also road lane marks and some other areas (blue box) because of edge detection. However, the initial change map detected by ConsimNet does not include this noisy object. Therefore, the edge map can be considered as a reference of the initial change map to improve the accuracy of the result.

2.4. Evaluation Metrics

In this study, the accuracy of the method was defined using three different performance metrics [5,20,21].

Recall:
$$Re = \frac{T_{\rm P}}{T_{\rm P} + F_{\rm N}}$$
 (6)

Precision :
$$\Pr = \frac{T_{\rm P}}{T_{\rm P} + F_{\rm P}}$$
. (7)

$$F - \text{measure}: F = 2\frac{P_r * R_e}{P_r + R_e}$$
(8)

where T_P is the number of true positives (i.e., the cases that were correctly classified), F_P is the number of false positives (i.e., the negative pixels that were incorrectly classified as positive pixels), and F_N is the number of false negatives (i.e., the positive pixels that were incorrectly classified as negative).

3. Experiment

3.1. Study Area and Devices

The object of the survey was the Deokyang Bridge (Figure 4) in Yeosu City, Korea. Deokyang Bridge is 530 m long and 25 m wide. To detect the surface changes, data were taken at different times

by a Drone Phantom 4 RTK (Table 1). The first recording was conducted on 11 January 2019, and the second was on 17 April 2019. An orthomosaic whose average ground resolution was 14 mm/pixel was generated through photogrammetric processing. The bridge area was selected as the test area of the orthomosaic image. That area was divided into our computer-processable size, and the same area of two period images was selected to finally generate 163 comparison pairs.



Figure 4. Deokyang bridge area.

Table 1. Phantom 4 RTK and mounted camera specifications (DJI Corporation).

Phantom 4 RTK Parameters					
Maximum speed	um speed 58 km/h				
Flight time	30 mins				
Camera Parameters					
Lens	FOV 84°; 8.8 mm/24 mm				
Sensors	1" CMOS; Effective Pixels: 20M				
Image sizes	4864 × 3648 (4:3); 5472 × 3648 (3:2)				
Gimbal	Stabilization: 3-axis Pitch: -90° to $+30^{\circ}$				

3.2. Implementation Details

To train the proposed network, this study used a CDnet dataset [20,21]. This dataset has already been used in [9,22]. The CDnet dataset consists of 31 videos with 91,595 image pairs depicting indoor and outdoor scenes with pedestrians, boats, and trucks captured at different times. The dataset represents various challenges divided into categories such as dynamic backgrounds, camera jitter, shadow, night video, challenging weather, and internal object motion. A background image with no feature object was selected for the reference image at time t_0 , and other images were taken at time t_1 . A total of 91,595 image pairs were used for the training, comprising 73,276 pairs for the training set and 18,319 for the validation set. All images were scaled to 512×512 during the training. The proposed Siamese network was implemented using the PyTorch framework [23]. In the training procedure, the learning rate was set to 0.00001, and the weight decay and momentum were set to 0.00005 and 0.9, respectively. The batch size was set to 32. The entire process of training, testing, and checking the results was performed in Python on a PyTorch platform [23] running a Linux 18.04 operating system. The training hardware used the NVidia Titan Xp graphics processing unit.

4. Results

To determine whether the method was a true detection method, 163 pairs of small images were used to preliminarily determine how many locations the detection method identified, that is, the

number of correct locations and the number of incorrect locations. The results are shown in Table 2 and Figure 5.



Table 2. Change detection results.

(d) False negative

Figure 5. Change detection results: (a) True negative: No changed area and no detection; (b) True positive: Existing damage area and true detection; (c) False positive: No damage area and false detection; (d) False negative: Existing damage area and no detection.

Given the results above, we can see that out of the 163 images, 138 images provided good results, equivalent to 84.7%; and 25 images provided incorrect results, equivalent to 15.3%. Thus, these results reflect the accuracy of the method.

For a more detailed evaluation, seven image pairs were used for testing. An image-to-image registration process was used to ensure that the image pairs were matched and located. The results are shown in Figures 6–12, which represent tests 1 to 7, respectively. Each figure contains smaller images, labeled (a)–(f). Image (a) is the image at time t_0 captured on 11 January 2019 and image (b) is the image at time t_1 captured on 17 April 2019. Image (c) is the initial unimproved result, and image (d) is the blended result between the image at time t_1 and the initial results. Images (e) and (f) are the preliminary differences between the two input images. Image (g) is the result after improving, and image (h) is the blended result between the improved result and the image at time t_1 .



Figure 6. Change detection task—Test 1: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the improved result and the image at time t_1 .



Figure 7. Change detection task—Test 2: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the improved result and the image at time t_1 .



Figure 8. Change detection task—Test 3: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the improved result and the image at time t_1 .



Figure 9. Change detection task—Test 4: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the improved result and the image at time t_1 .

As can be seen in images (c), (d), (g), and (h) of Figures 6–12, there are various colors in the detected area, including green, yellow, orange, and red. This is a result of the different distances between the two feature pairs, which were calculated by the Euclidean distance and contrastive loss functions. Change distance images between the feature pairs were enhanced with a rainbow color map for visualization contrast.



Figure 10. Change detection task—Test 5: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the image at time t_1 .



Figure 11. Change detection task—Test 6: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the image at time t_1 .

As seen in the blended image (d), all potholes were correctly detected; however, the extents of the detected area and the real changed area in image t_1 were unequal. Figure 6d—Test 1, Figure 7d—Test 2, Figure 9d—Test 4, and Figure 11d—Test 6 show that, before improvement, the extents of the detected areas were smaller than those of the real damaged areas. However, looking at Figures 6h, 7h, 9h and 11h, after improvement, the full extent of the damaged areas could be detected. The effectiveness of the method is shown numerically in Table 3.



Figure 12. Change detection task—Test 7: (**a**) image at time t_0 ; (**b**) image at time t_1 ; (**c**) the initial unimproved result; (**d**) the blended result between the image at time t_1 and the initial results; (**e**) and (**f**) preliminary differences between the two input images; (**g**) result after improving; (**h**) the blended result between the image at time t_1 .

Table 3. Number of changed areas detected by the proposed method. Pr: precision; Re: recall; F: F-measure.

Image Pairs -	ConsimNet			ConsimNet with Improvement		
	Pr (%)	Re (%)	F (%)	Pr (%)	Re (%)	F (%)
Test 1	67.88	1.71	3.33	74.86	2.30	4.47
Test 2	51.12	0.8	1.58	78.74	1.56	3.05
Test 3	65.19	0.1	0.2	76.24	0.29	0.58
Test 4	20.86	0.45	0.87	83.20	2.11	4.11
Test 5	87.21	0.29	0.58	88.18	0.39	0.77
Test 6	72.71	0.57	1.13	70.3	0.86	1.71
Test 7	58.43	0.1	0.2	79.39	0.23	0.47
Average	60.49	0.57	1.13	78.7	1.10	2.17

To evaluate the performance of the proposed method, the precision (*Pr*) was calculated as the division of the correctly classified changed area (T_p , true positive) by the sum of the correctly classified changed area (T_p , false positive). As (*Re*) was the result of the correctly classified changed area (T_p) divided by the sum of the correctly classified changed area (T_p) and correctly classified unchanged area (F_p), the *F*-measure rate (*F*) was the harmonic mean of the precision (*Pr*) and the recall (*Re*).

Seven image pairs were used for the test; in general, the values of *Pr*, *Re*, and *F* for the results after improvement were improved. Across all seven tests, before the adjustment, the average value of *Pr* was 60.49%, *Re* was 0.57%, and *F* was 1.13%; after adjustment, the values of *Pr*, *Re*, and *F* were 78.7%, 1.10%, and 2.17%, respectively.

5. Conclusions

In this study, a change detection method based on a convolutional Siamese network was introduced for UAV-obtained road surface images. The feature pairs of two UAV images taken at different times were extracted by the convolutional Siamese network. Then, the distance between the features was generated to detect changes between image pairs. The contrastive loss was applied to push changed pairs apart and pull unchanged pairs together. Finally, edge detection was used to obtain the boundaries of changed areas, and based on these boundaries, it was possible to adjust the detected area in the initial change map. This method can help to warn managers experts about road surface conditions. The method not only determined the location of the changing area, it also ensured that the full extent of the changing area was detected. Once the defect was detected, countable values like area or position of the defect could be obtained based on the pixel size. If a classification by damage type is added in the future, we could further develop this method into a pavement management system along with damage location.

However, there are still some difficulties caused by noise-generating objects, as some unwanted objects can still be detected and cause confusion. The most unwanted noise in this research is caused by severe shadows. Although shadows themselves were not detected as changes in the images, the actual defects lying beneath these shadows can remain undetected. If the view geometry of the cameras at different times is significantly different (e.g., with a difference of 30° or more, the detection rate is lowered. Standing water on roads also causes errors. Therefore, it is recommended to shoot the road almost vertically when the sun is high, or when the weather is slightly cloudy.

Determining the type and size of the detected damage depends on the ground spatial resolution of the images. That is, in high-resolution images, it is possible to detect changes of minute linear cracks, etc.; however, in centimeter-level low-resolution images, it is possible to detect the presence of potholes on roads. It is necessary to use sub-millimeter resolution images to detect minute linear cracks and crack changes at the millimeter level. In the future, we will study the change detection of small features due to seasonal variation or deterioration.

Author Contributions: D.H. conceived the idea, designed the framework; T.L.N. and D.H. carried out the experiments and analyzed the data; T.L.N. and D.H. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), which is funded by the Ministry of Education (NRF-2016R1A6A3A11930130).

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Li, F.; Fang, F.; Zhang, G. Unsupervised change detection in SAR images using curvelet and L1-norm based soft segmentation. *Int. J. Remote. Sens.* **2016**, *37*, 3232–3254. [CrossRef]
- Radke, R.; Andra, S.; Al-Kofahi, O.; Roysam, B. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Process.* 2005, 14, 294–307. [CrossRef] [PubMed]
- 3. Lu, D.; Mausel, P.; Brondizio, E.; Morán, E. Change detection techniques. *Int. J. Remote. Sens.* 2004, 25, 2365–2401. [CrossRef]
- 4. Im, J.; Jensen, J.R.; Tullis, J.A. Object-based change detection using correlation image analysis and image segmentation. *Int. J. Remote. Sens.* **2008**, *29*, 399–423. [CrossRef]
- 5. Makuti, S.; Nex, F.; Yang, M.Y. MULTI-TEMPORAL CLASSIFICATION AND CHANGE DETECTION USING UAV IMAGES. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, 42, 651–658. [CrossRef]
- 6. Yuhaniz, S.; Vladimirova, T. An onboard automatic change detection system for disaster monitoring. *Int. J. Remote. Sens.* **2009**, *30*, 6121–6139. [CrossRef]
- 7. Zhong, J.; Wang, R. Multi-temporal remote sensing change detection based on independent component analysis. *Int. J. Remote. Sens.* 2006, 27, 2055–2061. [CrossRef]
- 8. Alcantarilla, P.F.; Stent, S.; Ros, G.; Arroyo, R.; Gherardi, R. Street-view change detection with deconvolutional networks. *Auton. Robot.* **2018**, *42*, 1301–1322. [CrossRef]
- 9. Guo, E.; Fu, X.; Zhu, J.; Deng, M.; Liu, Y.; Zhu, Q.; Li, H. Learning to Measure Change: Fully Convolutional Siamese Metric Networks for Scene Change Detection. *arXiv* **2018**, arXiv:1810.09111.
- 10. Sakurada, K.; Okatani, T. Change Detection from a Street Image Pair using CNN Features and Superpixel Segmentation. In Proceedings of the 2015 British Machine Vision Conference, Swansea, UK, 7–10 September 2015.
- 11. Nemmour, H.; Chibani, Y. Fuzzy neural network architecture for change detection in remotely sensed imagery. *Int. J. Remote. Sens.* **2006**, *27*, 705–717. [CrossRef]
- 12. Liu, X.; Lathrop, R.G. Urban change detection based on an artificial neural network. *Int. J. Remote. Sens.* **2002**, *23*, 2513–2518. [CrossRef]

- 13. Wang, W.; Hall-Beyer, M.; Wu, C.; Fang, W.; Nsengiyumva, W. Uncertainty Problems in Image Change Detection. *Sustain.* **2019**, *12*, 274. [CrossRef]
- 14. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial Images. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 1845–1849. [CrossRef]
- Song, F.; Dan, T.; Yu, R.; Kun, Y.; Yang, Y.; Chen, W.; Gao, X.; Ong, S.-H. Small UAV-based multi-temporal change detection for monitoring cultivated land cover changes in mountainous terrain. *Remote. Sens. Lett.* 2019, *10*, 573–582. [CrossRef]
- 16. Shi, J.; Wang, J.; Xu, Y. OBJECT-BASED CHANGE DETECTION USING GEOREFERENCED UAV IMAGES. ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 2012, 38, 177–182. [CrossRef]
- 17. Altuntas, C. Urban Area Change Detection Using Time Series Aerial Images. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, *42*, 29–34. [CrossRef]
- Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006.
- Bromley, J.; Bentz, J.W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Sackinger, E.; Shah, R. SIGNATURE VERIFICATION USING A "SIAMESE" TIME DELAY NEURAL NETWORK. *Int. J. Pattern Recognit. Artif. Intell.* 1993, 7, 669–688. [CrossRef]
- 20. Goyette, N.; Jodoin, P.-M.; Porikli, F.; Konrad, J.; Ishwar, P. Changedetection. net: A new change detection benchmark dataset. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012.
- 21. Wang, Y.; Jodoin, P.-M.; Porikli, F.; Konrad, J.; Benezeth, Y.; Ishwar, P. CDnet 2014: An expanded change detection benchmark dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014.
- 22. Lee, S.-H.; Lee, G.-C.; Yoo, J.; Kwon, S. WisenetMD: Motion Detection Using Dynamic Background Region Analysis. *Symmetry* **2019**, *11*, 621. [CrossRef]
- 23. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch. In Proceedings of the 31st on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).